



Szendrei Rudolf

PhD hallgató
ELTE Informatikai Kar
swap@inf.elte.hu

Dr. Elek István

Térképtudományi és Geoinformatikai Tanszék
ELTE Informatikai Kar
elek@map.elte.hu

Dr. Fekete István

Algoritmusok és Alkalmazásai Tanszék
ELTE Informatikai Kar
fekete@inf.elte.hu

Dr. Márton Mátyás

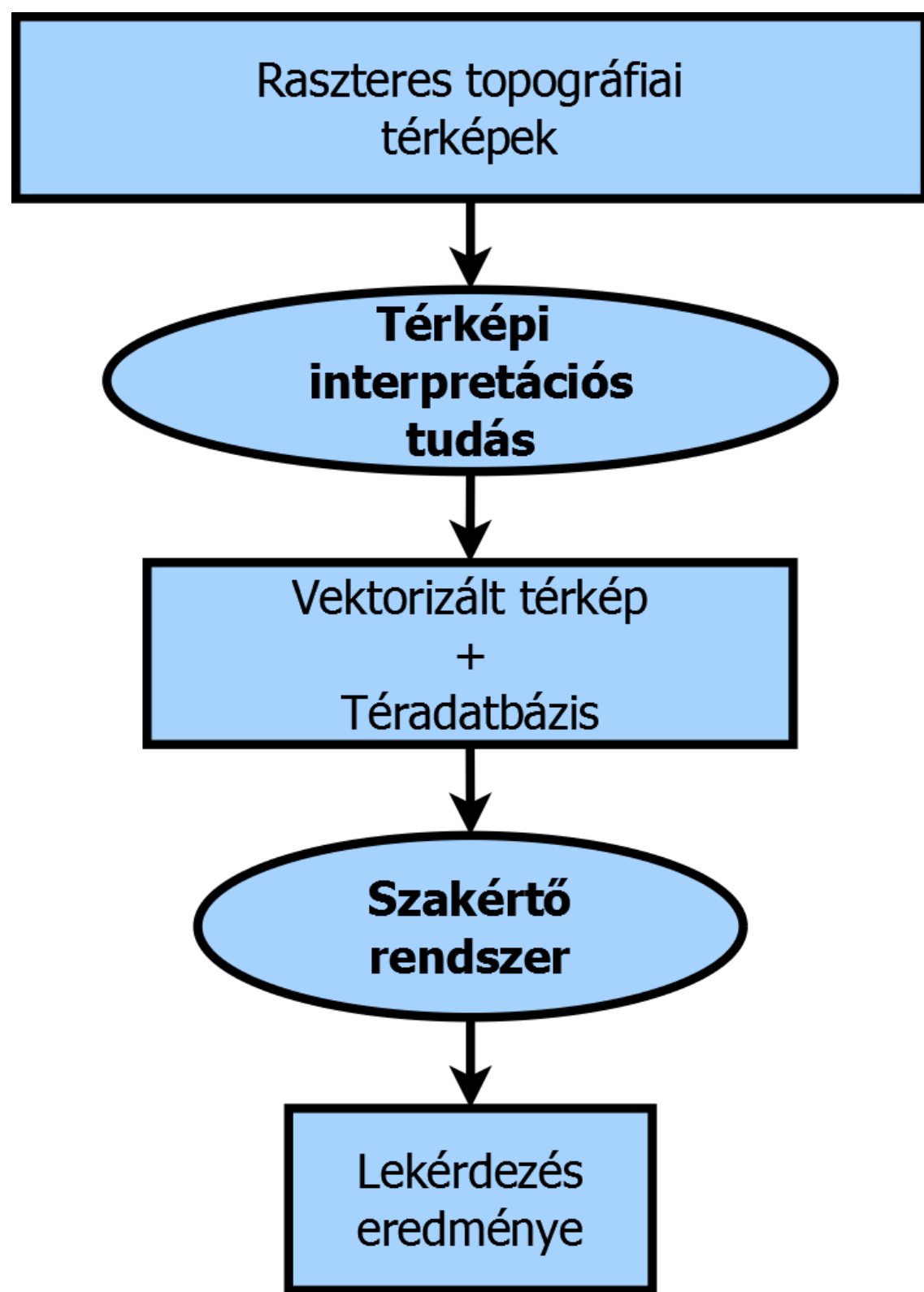
Térképtudományi és Geoinformatikai Tanszék
ELTE Informatikai Kar
matyi@map.elte.hu

1 Kivonat

A térinformatikában és a hétköznapi térképhasználatban napjainkra a vektoros térképek használata válik jellemzővé, mivel azok a térinformatikai adatbázisokból könnyen elérhetők és térbeli lekérdezéseket tudunk rajtuk végezni. A régebbi papírtérképek „kézzel” végzett vektorizálása azonban hosszadalmas és költséges eljárás, ezért célunk ennek a folyamatnak az automatizálása. Korábbi IRIS (Intelligens raszterkép interpretáló rendszer) projektünkben az egyes térképi jelkulcselem típusokhoz különálló felismerő algoritmusok készültek. A térkép jó minőségű vektorizálásához azonban szükséges megértenünk a térképen látottakat, ismernünk a térképkészítési eljárást, a térképész szakemberek gondolkodásmódját. Ennek a megoldására a mesterséges intelligenciát hívtuk segítségül, ahonnan különféle heurisztikákat, következtetési szabályokat, valamint egy speciális gráf alapú tudásreprezentációt használtunk.

2 Bevezetés

A nyomtatott topográfiai térképek vektorizálását még napjainkban is főként „kézzel” végzik. A konverzió során létrejövő térinformatikai adatbázison különféle lekérdezések fogalmazhatók meg, melyekre alapozva több szakértői rendszert is készítették, például áradás modellezésére. Az ilyen rendszerek pontossága az adatok minőségén múlik. Egy raszteres térképtől a vektoros lekérdezésig tartó folyamat során a tudásábrázolást két nagy csoportra osztottuk. Az első csoportba a raszter-vektor konverzió tudás alapú algoritmusait soroltuk - melyekkel mi foglalkozunk -, míg a másodikba a vektoros modellen működő szakértői rendszerek tudás-elemeit helyeztük.

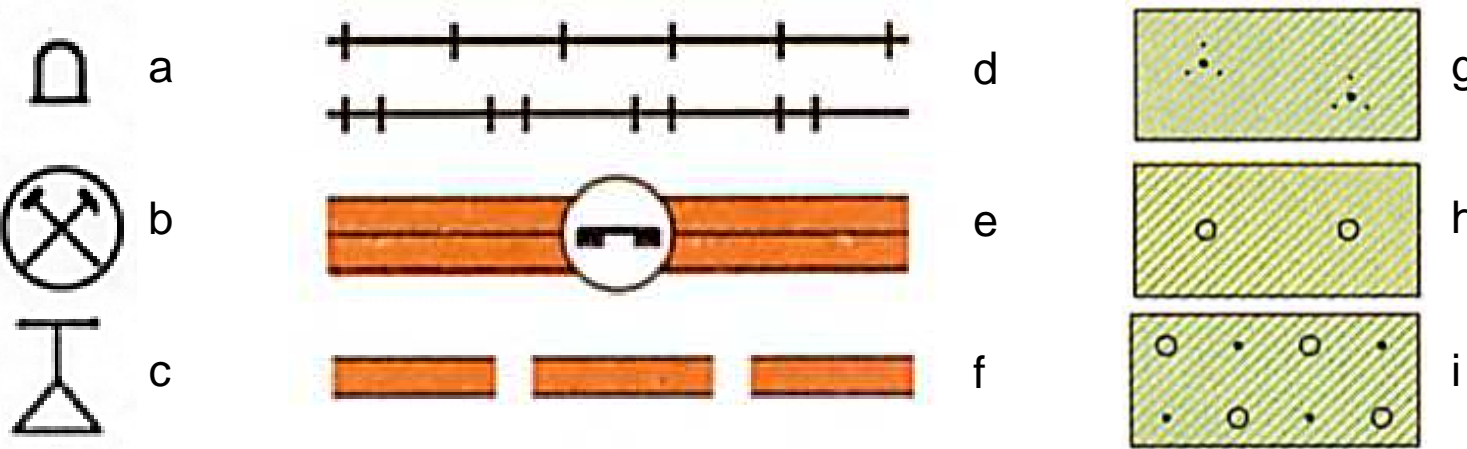


A térképen látottakat algoritmikus szempontból csoportosítottuk aszerint, hogy az egyes algoritmusok milyen térképi objektumtípusok felismerésére alkalmasak. A pontszerű objektumok felismerésére egy hatékony, lineáris idejű, heurisztikus, raszteres mintaillesztő algoritmust dolgoztunk ki, míg a felületek felismerésénél ennek az algoritmusnak egy módosított változatát használjuk fel. A vonalas objektumok felismerésére egy gráf-alapú felismerő eljárást mutatunk be, ahol az egyes vonalas elemek strukturális jellemzőit egy-egy gráf-topológiájával ábrázoljuk. A felismerési problémát véges determinisztikus automatákkal végzett gráfösszehasonlításra vezetjük vissza. A fent említett algoritmusokat egy térképi maszkreteg használatával kapcsoljuk össze logikailag. Mivel a térképi elemek a térképkészítéskor különálló rétegeken helyezkednek el mielőtt egymásra rajzolnák őket, ezért felhasználjuk azok keletkezési sorrendjét, takarási logikáját. Az egyes felismerési lépésekben a felismert objektumokat a maszkretegen töröljük, ezáltal a következő lépésekben csak a maradék adatokon dolgozunk. A kérdéses vektorizálatlan vonalas részeket, illetve az ismeretlen felületeket következtetési szabályok segítségével próbáljuk meg felismerni.

Megjegyezzük, hogy a nagyléptékű topográfiai térképeken látható információk közül csak a pontszerű és vonalas objektumok, valamint a felületek felismerésével foglalkozunk, és eltekintünk a számadatoktól, illetve a szöveges címkektől.

3 Térképi jelkulcs-elemek

A raszter vektor konverzió során a topográfiai térképek jelkulcs-elemeit három csoportba soroljuk. Pontszerűnek tekintjük azokat a kis ábrákat a térképen, amelyek egy konkrét helyen előforduló térbeli objektumot reprezentálnak. A vonalas csoportba soroljuk a szem számára vonalas grafikai elemeket, amelyek lehetnek pl. utak, vasutak, patakok, illetve határoló vonalak. A felületek csoportjába kerülnek azok a térképi részek, amelyek egy adott homogén színnel fedettek, vagy amelyeket egy ismétlődő, mozaikszerűen elhelyezett mintázat fed le.



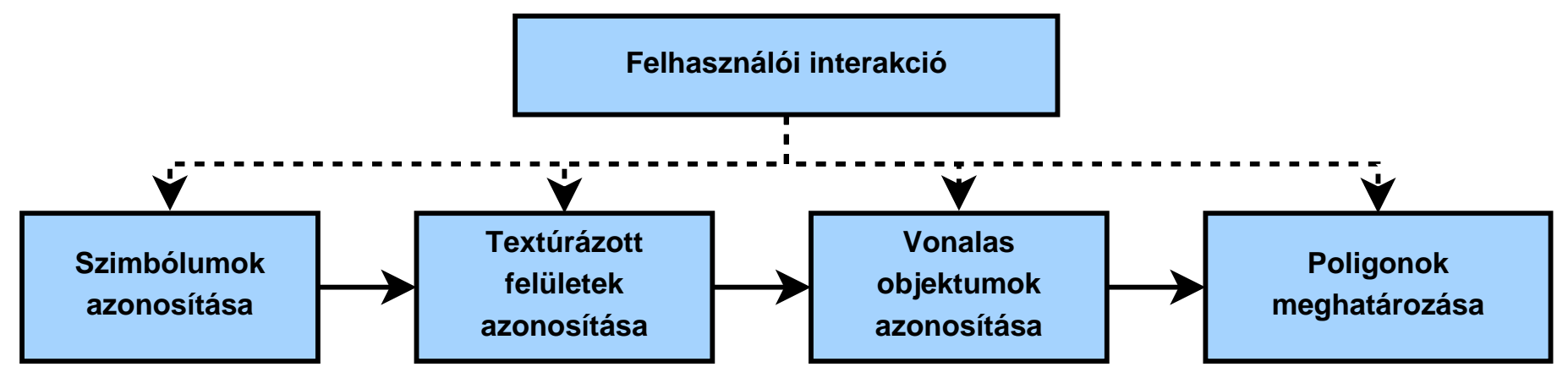
Pontszerű objektumok: a) szobor, b) tárna bejárat, c) meteorológiai állomás. Vonalas objektumok: d) vasút, e) autópálya, segélyhívóval, f) épülő műút. Felületi jelkulcs-elemek: g) bozótos, h) gyümölcsös, i) gyümölcsös, bokrokkal.

4 Tudásalapú megközelítés

A térképkészítés menetét vizsgálva a következő általános sorrendet állapítottuk meg:

1. Vízrajzi réteg elkészítése
2. Polygonok határvonalainak megrajzolása
3. Polygonok kifestése vagy textúrázása
4. Úthálózat megrajzolása
5. Pontszerű objektumok feltüntetése
6. Szöveges és számadatok felírása a térképre

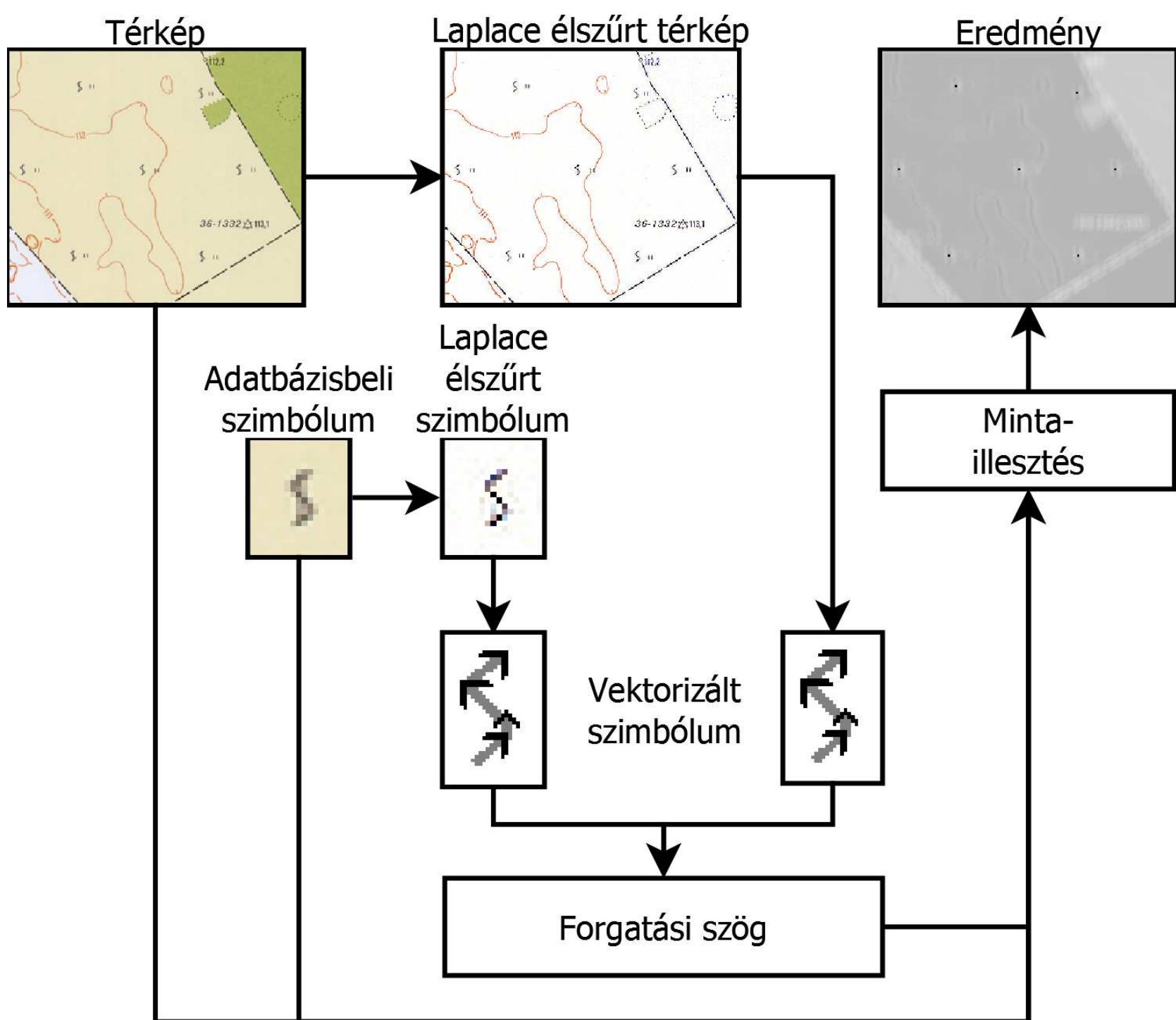
Mivel az egyes rétegek egymásra rajzolódnak, ezért gyakori problémát jelent a kitakarás, így kiindulásként az egyes jelkulcs-elemek felismerését a térképkészítés menetének fordított sorrendjében próbáljuk meg elvégezni. A gyakorlatban a textúrázott felületek sokszor zavaró tényezőként hatnak a vonalas adatok felismerésekor, ezért a felületi réteget a folyamatban előre vesszük, így ennek segítségével kiszűrhetjük a zavaró területeket a vonalas adatok felismerésénél. A folyamatot négy szakaszra bontottuk - megengedve eközben a felhasználónak, hogy beavatkozzon és korrigáljon az adatokon.



A folyamat során eltávolítjuk az egyes objektumok vektoros adatait és jelöljük az általuk fedett területeket egy különálló maszkretegen. Később ezek segítségével logikai szabályok fogalmazhatók meg, amelyekkel dönthetünk például akkor, ha egy folyó folytonosságát egy híd szakítja meg.

5 Pontszerű jelkulcs-elemek

Egy korábbi cikkünkben már részletesen bemutatunk egy - pontszerű jelkulcsi elemek textúra alapú felismerését végző - lineáris futási idejű algoritmust. Az ábrán az algoritmus folyamatábrája látható, amint egy adott jelkulcs-elemet próbál meg felismerni a térképen.



Az algoritmus egy pontszerű térképi jelnek csak a potenciális előfordulási helyeit vizsgálja meg, melyeket a térkép és a jelkulcs képének élszűrt változatából nyer ki egy Otsu-küszöbölés után. A kapott binarizált képnek csak azon pozícióira illetve végzünk raszteres mintaillesztést, ahol egyes érték szerepel. Mivel ez az algoritmus csak egyenként képes a pontszerű jelkulcs-elemeket felismerni, ezért készítettünk egy heurisztikát, amellyel egyszerre felismerhető az adatbázis valamennyi pontszerű jelkulcsi eleme.

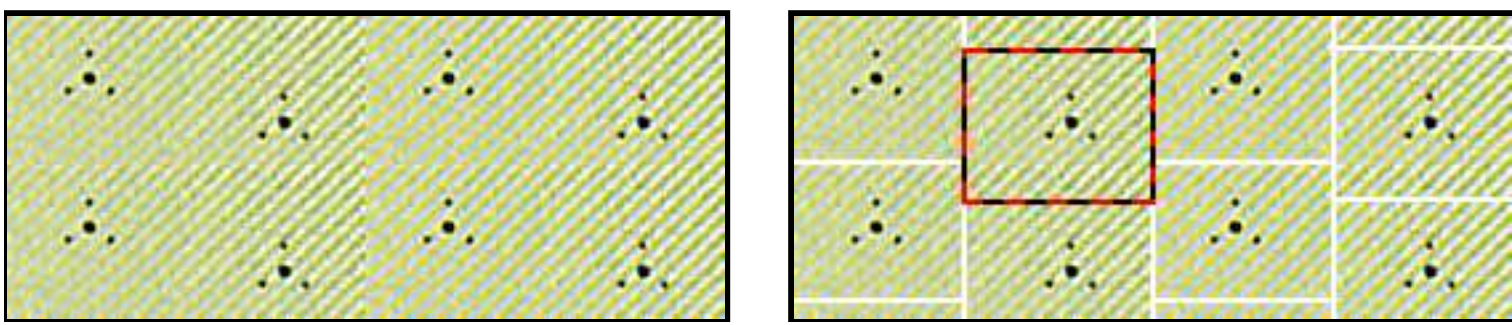
Első lépésként az adatbázis pontszerű jelkulcselemeinek raszteres képeire kiszámoljuk azok szórását és eszerint sorba rendezzük őket.

...	0.43	0.48	0.49	0.57	0.71	...
-----	------	------	------	------	------	-----

Mielőtt elkezdenénk az algoritmust futtatni, számítsuk ki a potenciális pontban az illesztendő jelkulcs-elem alá eső térképi terület szórását, majd keressük ki ezt módosított "bináris legközelebbi elem" kereséssel a "legközelebbi" jelkulcs-elemet. Ha ez a mintaillesztés során nem ismerhető fel, akkor "jobbra-balra lépegetéssel" keressünk tovább az elemek között egy bizonyos szóráseltérés megengedése mellett. A felismert jelkulcs-elem helyét és típusát egy vektoros fájlba exportáljuk külön térképi réteggént. Az eljárás során felismert szimbólumok által lefedett térképi területeket a maszkretegről eltávolítjuk az ezután következő algoritmusok hatékonyabbá tételének érdekében.

6 Textúrázott, felület alapú jelkulcs-elemek

A textúrázott felületek felismerésénél a pontszerű szimbólumok felismerését végző raszteres mintaillesztő algoritmus egy módosított változatát használjuk. Első lépésben megkeressük azt a legkisebb részét a jelkulcs-elem textúrájának, amely generálja azt (mozaikszerűen, hézag- és átlapolásmentesen lefedhető vele a textúra).

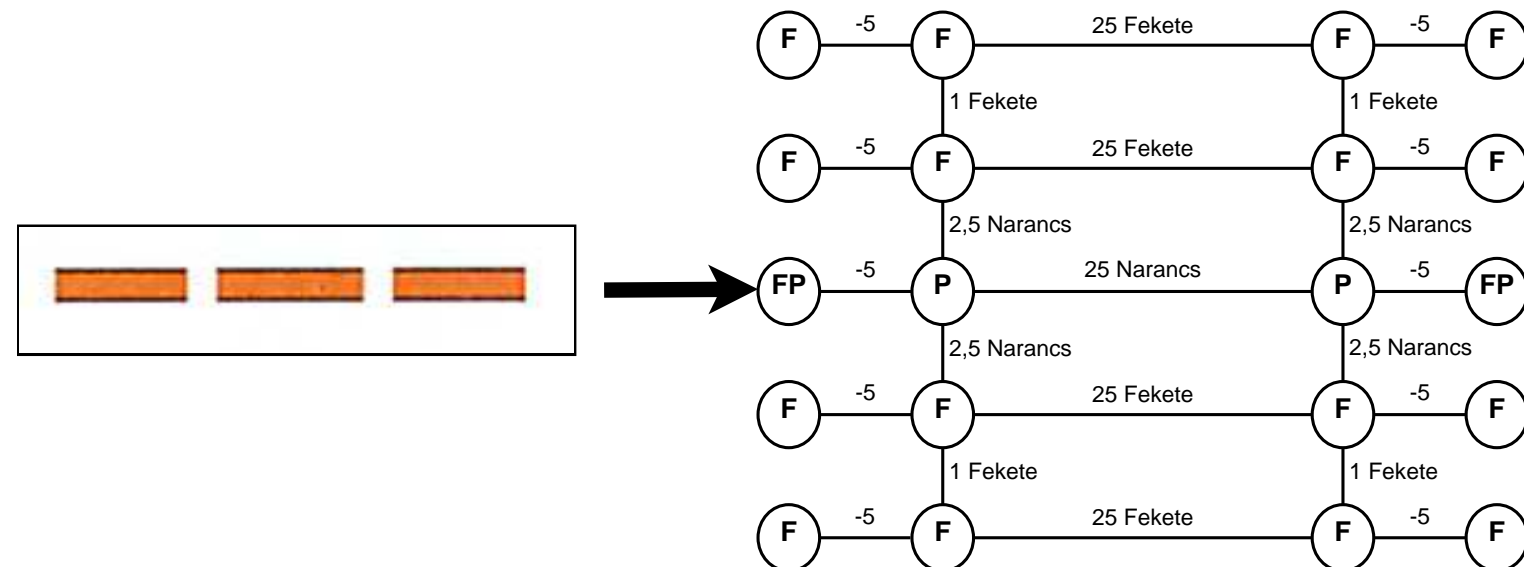


Cserjést ábrázoló textúra és annak kernelje

A textúrák felismerésénél először teljes mintákat keresünk, majd ezután a teljes mintákhoz kapcsolódó részleges mintákat. A kereséskor figyelembe vesszük a maszkreteget, így csak azokat a pixeleket vizsgáljuk, amelyeket nem távolított el a maszkreteg. A felismert minták alatti területekhez hozzárendeljük a nekik megfelelő típust és kimaszkoljuk a nekik megfelelő részeket. Végezetül a felületeket típusonként külön vektoros állományba mentjük el.

7 Vonalas jelkulcs-elemek

A vonalas jelkulcsi elemek felismerésekor a térkép élszűrt változatának Otsu-küszöbölt bináris képét próbáljuk meg vektorizálni. Ennek során nem vesszük figyelembe azokat a térképi részeket, amelyeket a maszkreteg szerint korábban már értelmeztünk. A vektorizálás során bejárjuk az élpixeleteket és vektorokká alakítjuk őket. Mivel a vonalas adatok jellegzetes topológiával rendelkeznek, ezért egy egyszerű bejárás nem elegendő ezek értelmezéséhez. A topológia értelmezéséhez a vonalak esetében az elágazási és végződési pontokat vizsgáljuk (hasonlóan az ujjlenyomat felismeréséhez). A mi esetünkben ezen speciális, szomszédos pontok bejárás szerinti távolságát vizsgáljuk. Az elágazási és végződési pontok kapcsolata és távolsága alapján megfigyelhetünk egy gráfot, ahol F jelöli az elágazást, P pedig az úton fekvő pontot. A gráfban az élekhez rendeljük a megfelelő távolságot, illetve a jelkulcs szerinti szint is. A ciklikusságot az egy szomszédú elágazás-csúcsok jelölik.



A gráf alapú topológia vizsgálatánál a leghatékonyabb módszer az, ha az adott pontból induló bejárásnál egy véges determinisztikus automatát készítünk a gráfokhoz. Ezek az automaták jól ismertek a programozási nyelvek esetében használatos fordítóprogramok tokenizálójában, így követhetjük azok analógiáját. Ahogy a megelőző két lépésben is tettük, a felismert objektumok által takart részeket ebben a lépésben is kimaszkoljuk a maszkretegben.

8 Polygonok felismerése

A megelőző lépésekben megpróbáltuk felismerni a térképi elemeket, azonban néhányuk nem ismerhető fel, csak megfelelő következtetési szabályok alkalmazásával. A gyakorlatban sokszor előfordul például az az eset, hogy egy ház poligonja egy rózsaszín színnel kitöltött, vékony fekete kerettel határolt objektumként látható a térképen. Ekkor a felület a kis mérete miatt nem ismerhető fel felületi objektumként. Vonalas objektumként szintén nem ismerhetjük fel az ilyen objektumokat, hiszen ezek vektoros modelljei egyszerű zárt görbék (ezek kivételes esetekben egymáshoz érnek a térképen). A fentebb említett algoritmusok által fel nem ismert adatok tehát poligonok, melyekre külön módszert kell kidolgoznunk. Ennek egy lehetséges módja az *AB-szabály* alkalmazása.

A	Narancs-vörös	Narancs	Tengerkék
B	Fehér	Fehér	Egyéb
Következtetés	Földszintes ház	2-4 szintes ház	Vízfelület

Egy *AB-szabály* segítségével egy vektoros modellbeli vonalszakasztól el tudjuk dönteni optimális esetben, hogy milyen típusú objektum található az egyik, illetve a másik oldalán. Mivel a vonalszakaszok vektorokként vannak ábrázolva a térinformatikai adatbázisokban, ezért megkülönböztethetjük a szakasz két oldalát, elnevezve ezeket A, illetve B oldalnak. A két oldalnak, mint területnek a színét, illetve típusát ismerhetjük, mint tulajdonságot. A mi esetünkben az egyszerűség kedvéért csak a területek színét használjuk. A fenti példában jól látható, hogy egy ház poligonjának határolóvonalát, mint vonalszegmens-sorozatot miként nyilváníthatjuk ház típusúnak, annak alapján, hogy a poligon belseje rózsaszínnel van kifestve.

9 Konklúzió

A tudásalapú térképi raszter-vektor konverzió négy egymásra épülő heurisztikus algoritmusát készítettük el, amelyeket egy térképi maszkreteggel kapcsolunk össze. Az algoritmusok az általuk felismert térképi objektumokat különálló vektoros adatállományokban helyezik el. A vektorizálás során az algoritmusok egy olyan tudásbázist használnak fel, amely tartalmazza a pontszerű jelkulcsi elemek raszteres képeit, szórásukat és metaadataikat, a vonalas elemek topológiáját leíró gráfokat és azok véges determinisztikus automatáit, valamint a poligonok felismeréséhez és kategorizálásához szükséges következtetési szabályrendszert.

A jövőben a következtetési szabályrendszert szeretnénk kibővíteni a vonalas objektumok anomáliáinak kezelésére vonatkozó szabályokkal is (pl. felüljáró által vizuálisan megszakított út). A vízfelületek is további gondolkodást igényelnek, mivel itt az anomáliák kitakarás esetében együttesen érintik a vonalas és felületi objektumokat (pl. híd egy folyó felett).